APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/657,770	09/08/2003	Poon Fung	IAM-001	3952

7590	05/24/2006

Poon Fung
20375 Via Volante
Cupertino, CA 95014

EXAMINER
SCHELL, JOSEPH O

ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 05/24/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/657,770	FUNG ET AL.
	Examiner	Art Unit	
	Joseph Schell	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on _08 September 2003_.
2a) ☐ This action is **FINAL.** 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) _1-22_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) _1-22_ is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on _08 September 2003_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

    1. ☐ Certified copies of the priority documents have been received.

    2. ☐ Certified copies of the priority documents have been received in Application No. _____.

    3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Detailed Action

Claims 1-22 have been examined.

Claims 1-22 have been rejected.

Claim Objections

1.   In claim 16 line 13, it is believed that the claim should read "collected data **from** all said application components" unless Applicant is claiming that data is recorded within the code of the applicant components being traced. Such a limitation would require further enablement within the specification.

2.   In claim 20 line 6, the last element of the claim needs to be preceeded by "and" or "or" to define the necessity of all elements or just some elements. From page 8 line 29 through page 9 line 2 of the specification, the examiner assumes all elements of the claim are required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3.   Claim 6 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 6 recites the limitation "said unique value" in the first

line of the claim. There is insufficient antecedent basis for this limitation in the claim.

Examiner assumes that claim 6 is instead dependent on claim 5 which does recite this

limitation.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4.   Claim 22 is rejected under 35 U.S.C. 101 for not being limited to statutory

embodiments. A program product claim needs to be stated in such a way that it is

limited to only tangible, material embodiments, and avoids abstract code or code carrier

wave embodiments.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5.   Claims 1-5, 7-8, 11-18 and 21-22 are rejected under 35 U.S.C. 102(b) as being

anticipated by Linnermark (US Patent 5,594,904).

6.   As per claim 1, Linnermark ('904) discloses a computer implemented method for

tracing an execution path through application components distributed over at least one

computer in a data processing system (column 3 lines 40-45), comprising the steps of:

generating a tracing token (column 6 lines 33-34), activating data collection

(column 6 lines 8-14 and column 11 lines 19-26), and relaying said tracing token in one

of said application components distributed over at least one computer (column 6 lines

33-46); and

detecting said tracing token (column 6 lines 33-36), activating data collection

(column 6 line 54 through column 7 line 2), and relaying said tracing token in at least

one of said application components (column 6 lines 33-46 and as shown in figure 4, the

message with key moves from one lock to the next as execution progresses).

7.   As per claim 2, Linnermark ('904) discloses the method of claim 1, wherein said

application components include at least one of application programs (column 5 lines 50-

54), plug-in programs, program extensions, static library programs, dynamic library

programs, program scripts, and individual modules, or subroutines, or functions, or

procedures, or methods thereof.

8.   As per claim 3, Linnermark ('904) discloses the method of claim 1, wherein said

application components reside in the same computer, or are distributed over multiple

computers interconnected by at least one of a local area network, a wide-area network,

a wireless network, or the Internet (column 7 lines 7-12).

9.   As per claim 4, Linnermark ('904) discloses the method of claim 1, wherein said

generating a tracing token is controlled by a start rule (column 5 line 66 through column

6 line 2) and a stop rule (column 12 lines 57-61 and column 13 lines 63-67)

10.   As per claim 5, Linnermark ('904) discloses the method of claim 1, wherein said

tracing token is a unique value in said data processing system (column 12 lines 5-9 and

column 12 lines 15-18, each daemon is uniquely accessed and a key defines a set of

daemons to be accessed, thus each key is unique to a set).

11.   As per claim 7, Linnermark ('904) discloses the method of claim 1, wherein said

relaying of said tracing token is accomplished through at least one of reading and

writing of a shared memory, attachment to an inter-process message (column 6 lines

33-34), and insertion into a data stream.

12.   As per claim 8, Linnermark ('904) discloses the method of claim 7, wherein said

attachment to an inter-process message includes at least one of a non-intrusive

modification of the message header, a non-intrusive modification of the message body,

or appending to the message (column 6 lines 33-34).

13.   As per claim 11, Linnermark ('904) discloses the method of claim 1, wherein said

tracing an execution path through application components takes place in at least one of

system production time (column 10 lines 63-66 and column 17 lines 59-60), system

development time, system integration time, or system testing time of said data

processing system.

14.   As per claim 12, Linnermark ('904) discloses the method of claim 1, wherein said

generating a tracing token is automatically triggered by detection of a system event

(column 5 line 66 through column 6 line 2).

15.   As per claim 13, Linnermark ('904) discloses the method of claim 12, wherein

said detection of a system event is dynamically activated and deactivated (column 5

lines 60-64, in the "if (ON)" statement, ON is a variable determining whether detection is

activated).

16.   As per claim 14, Linnermark ('904) discloses the method of claim 13, wherein

said dynamic activation and deactivation of said detection of a system event are

controlled internally in said one of said application components, or externally to said one

of said application components (column 5 lines 47-50, the event monitoring is done by a

subprogram of the operating system and kernel).

17.   As per claim 15, Linnermark ('904) discloses the method of claim 12, wherein

said system event includes at least one of an event relating to system and application

performance (column 3 lines 34-39), system and application availability, system and

application troubleshooting, or system and application security.

18.   As per claim 16, Linnermark ('904) discloses a computer implemented method

for tracing an execution path through application components distributed over at least

one computer in a data processing system, comprising the steps of:

        installing monitoring code in a first plurality of said application components for

detecting system events (column 5 lines 47-50);

        installing monitoring code in a second plurality of application components for

detecting tracing tokens (column 5 lines 50-54);

        detecting one of said system events (column 6 lines 1-2), activating data

collection (column 6 lines 1-2), and generating and relaying one of said tracing tokens in

one of said first plurality of said application components (column 6 lines 33-34, the key

is relayed as the program executes);

        detecting and relaying said generated one of said tracing tokens to activate data

collection in at least one of said plurality of said application components along said

execution path through said application components in said data processing system

(column 6 lines 40-46 and as shown in Figure 4); and

        recording said collected data in all said application components along said

execution path through said application components in said data processing system

(column 6 lines 47-49).

19.   As per claim 17, Linnermark ('904) discloses the method of claim 16, wherein

said first plurality of said application components includes entry points for client

enquiries to an n-tier client/server distributed computing system (column 7 lines 15-32

describe a system of 2 communicating PBXs. Column 7 lines 44-49, daemons are

inserted at the server entry point, as shown in corresponding figure 5, element 54).

20.   As per claim 18, Linnermark ('904) discloses the method of claim 16, further

comprising the step of providing storage means for said collected data for all said

application components along said execution path through said application components

in said data processing system (column 6 lines 47-49).

21.   As per claim 21, Linnermark ('904) discloses the method of claim 16, wherein

said monitoring code is installed in at least one of the source code, the object code, the

executable code, and the application extension mechanism of said application

components (column 5 lines 50-54).

22.   As per claim 22, Linnermark ('904) discloses a computer program product

comprising:

        computer usable medium having computer readable program code means

embodied therein for tracing an execution path through application components

distributed over at least one computer in a data processing system (as shown by Figure

5, trace points are distributed throughout the networked system), the computer readable

program code means in said computer program product comprising:

computer readable program code means for installing monitoring code in a first

plurality of said application components for detecting system events (column 5 lines 47-

50);

computer readable program code mans for installing monitoring code in a second

plurality of application components for detecting tracing tokens (column 5 lines 50-54);

computer readable program code means for detecting one of said system events

(column 6 lines 1-2), activating data collection (column 6 lines 1-2), and generating and

replaying one of said tracing tokens in one of said first plurality of said application

components (column 6 lines 33-34, the key is relayed as the program executes);

computer readable program code means for detecting and relaying said

generated one of said tracing tokens to activate data collection in at least one of said

plurality of said application components along said execution path through said

application components in said data processing system (column 6 lines 40-406 and as

shown in Figure 4); and

computer readable program code means for recording said collected data in all

said application components along said execution path through said application

components in said data processing system (column 6 lines 47-49).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

23.   Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over

Linnermark ('904) in view of Mac OS X in a Nutshell.

Linnermark ('904) discloses the method of claim 5. Linnermark ('904) does not explicitly

disclose the method wherein the unique value includes a universally unique identifier.

Mac OS X in a Nutshell teaches general information about the uuidgen Unix command.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to use UUIDs within the system disclosed by Linnermark ('904). This use of UUIDs

would have been obvious because a UUID is guaranteed to be unique (Mac OS X in a

Nutshell), as is required of the key.

24.   Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over

Linnermark ('904) in view of Programming Web Services with SOAP.

Linnermark ('904) discloses the method of claim 8. Linnermark ('904) does not explicitly

disclose the system wherein said non-intrusive modification of the message header

includes at least one of adding a new field in a SOAP transport header over HTTP, or

adding a new property in a SOAP transport header over JMS.

Programming Web Services with SOAP teaches about the use of SOAP for

communications.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to modify the system disclosed by Linnermark ('904) such that messaging is done using

XML messages with a SOAP structure. This modification would have been obvious

because XML allows systems to communication regardless of operation systems or

other technical differences, while SOAP provides a standard way to structure XML

messages (Programming Web Services with SOAP, end of section 2.1.1). It would

have been further obvious to append the key onto a message by adding a new field in a

SOAP transport header. This would have been obvious because the SOAP message

header contains information relevant to how a message is to be processed, including

routing and delivery settings, authentication or authorization, and transaction contexts

(Programming Web Services with SOAP, top of section 2.2) and the key would the

classified as authentication information as it is compared with trace daemon locks.

25.   Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over

Linnermark ('904) in view of Microsoft SQL Server 2000 Unleashed.

Linnermark ('904) discloses the method of claim 8. Linnermark ('904) does not

expressly disclose the system wherein said non-intrusive modification of the message

body includes the insertion of a comment in an SQL statement.

Microsoft SQL Server 2000 Unleashed teaches the use of Microsoft's SQL Server and

related SQL features.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to modify the system disclosed by Linnermark ('904) such that the various servers of the

system run SQL Server Database. This modification would have been obvious because

SQL Server Database provides reliable storage, rapid data access, consistent data

access, and enforces data-integrity (Microsoft SQL Server 2000, Welcome to Microsoft

SQL Server chapter, SQL Server 2000 Components and Features section).

It would be further obvious to send messages between the SQL Server 2000 systems.

This would have been obvious because messages of a specified severity level can be

sent to another server for processing (Microsoft SQL Server 2000, SQL Server

Administration chapter, SQL Server Scheduling and Notification section, Event

Forwarding subsection).

It would have finally been obvious to insert a comment in an SQL statement when sending a message. This would have been obvious because it can help a user understand code (Microsoft SQL Server 2000 Unleashed, section on comments within the Transact-SQL chapter, between Listing 26.15 and Listing 26.16).

26.   Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Linnermark ('904).

27.   As per claim 19, Linnermark ('904) discloses the method of claim 18. Linnermark ('904) discloses that one of the advantages of the invention is that the event tracing information requires reduced storage (column 3 lines 23-25). Linnermark ('904) does not expressly disclose the method wherein said storage means includes at least one of files, registries, relational databases, and object databases.

At the time of invention it would have been obvious to a person of ordinary skill in the art to modify the system disclosed by Linnermark ('904) such that data stored is organized into files. A file provides the advantages of allowing manipulation of the data as a whole, allowing easy access by providing it with a descriptive name and breaking it into multiple files, and easy access and use through an operating system. These advantages would be apparent to one of ordinary skill in the art, making the storage of data in a file form obvious.

28.   As per claim 20, Linnermark ('904) discloses the method of claim 16 wherein a

record of said colleted data includes application data from operation (column 11 lines

11-17). Linnermark ('904) does not expressly disclose the method wherein a record of

said collected data includes: the value of one of said tracing tokens; the value of a

computer's timestamp; the identifier of a computer; the identifier of one of said

application components; and the description of one of said system events.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to include the value of a tracing token with logged information. The system disclosed by

Linnermark ('904) allows multiple traces to run simultaneously (column 3 lines 23-25).

The system traces by following messages as they run through multiple components (as

shown by Figure 5, the trace daemons are distributed through different processor

tasks). With the use of multiple traces from multiple components, it would be beneficial

to be able to identify a particular trace from the events listed in a log.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to include timestamp information with logged information. The inability of a trace

system to run for an extended time is one of the limitations overcome by the system

disclosed by Linnermark ('904) (column 2 lines 47-50 and column 3 lines 12-18). When

running a trace for an extended period of time it would be useful to have timestamp

information for logged events so that they can be easily correlated with known events

that were not logged in an easily comparable log. Events like network outages or router

failures can be matched with the corresponding log event they influenced.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to include computer identifier information with logged information. The system

disclosed by Linnermark ('904) employs multiple computer systems in a networked

system (as shown in Figure 1). A person viewing trace logs may only wish to see

events corresponding to a specific computer system, for example a newly installed

system or a computer system that has been problematic in the past.

At the time of invention it would have been obvious to a person of ordinary skill in the art

to include application component identifier information with the logged information. The

system disclosed by Linnermark ('904) adds trace daemons to multiple programs

(column 6 lines 8-10, sited elements 40-42). It would be useful to log the program

component information associated with a log entry for the user viewing the trace log

because tracing may occur over several applications on the same or multiple computer

systems (column 5 lines 34-42).

At the time of invention it would have been obvious to a person of ordinary skill in the art

to include the description of a system event with the logged information. The system

disclosed by Linnermark ('904) performs a trace on the occurrence of one or more

events and the events may vary in their effects on the trace system (column 6 lines 4-8,

the use of the term "particular event" implies that a variety of events are detected but only a particular kind initiate a trace). The variety of detectable events make it useful for a user viewing the logged information to have access to information regarding what event initiated a trace.

Conclusion

The prior art made of record on accompanying PTO 892 form and not relied upon is considered pertinent to applicant's disclosure. Specifically, Lindsey (US Patent 5,896,536) teaches a system wherein trace-points throughout an application collect and send information according to a trace-point variable, Frysinger (US Patent Application Publication 2004/0060043) teaches a system that performs code instrumentation that can be activated or deactivated depending on predetermined operational limits, and Diec (US Patent 6083281) teaches a distributed application with object tracing capabilities.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joseph Schell whose telephone number is (571) 272-8186. The examiner can normally be reached on Monday through Friday 9AM-4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Scott Baderman can be reached on (571) 272-3644. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

JS

SCOTT BADERMAN
SUPERVISORY PATENT EXAMINER